Amendments to the Specification:

Please replace the paragraph beginning at page 5, line 23 with the following paragraph:

Non-Volatile Memory circuit 20 is connected to a ~~KVG-68~~ KGV-68 encryption device 24 which allows Non-Volatile Memory circuit 20 to load a crypto key with its corresponding check word into the encryption device 24. The encryption device 24 is connected to a telemeter transmitter 26 which transmits encrypted telemetry data from encryption device 24 to a ground station, a ship, an aircraft or other receiving station.

Please replace the paragraph beginning at page 7, line 15 with the following paragraph:

During initialization the /VAR_REQ output of FIG. 3A from microcontroller 32 is set high since this signal is an active low signal. When the /VAR_REQ output signal of FIG. 3A is active low the encryption device 22 serially outputs the crypto key data which is illustrated in FIG. 3B and clock which is illustrated in FIG. 3C to microcontroller 32.

Please replace the paragraph beginning at page 7, line 18 with the following paragraph:

At this time it should be noted that the computer software program for microcontroller 32 is adapted for processing two KGV-68 encryption units although only one is illustrated in FIG. 1. In a security upgrade configuration the software operates in a manner which allows two KGV-68 encryption units to be loaded with a crypto key and its corresponding check word. It should be noted that while FIG. 1 only shows one ~~KVG-68~~ KGV-68, the non-volatile memory comprising the present invention may be easily modified to accommodate two ~~KVG-68~~ KGV-68 encryption units.

Please replace the paragraph beginning at page 9, line 13 with the following paragraph:

During program step 70 transmitter 26 is disabled by microcontroller 32 to prevent possible transmission of the crypto key. During program step 72 the /VAR_REQ output from microcontroller 32 is set low to request the check word from key loader 22. During program step 74 the check word is loaded into the ~~EEPROM of~~ microcontroller 32. During program step 76, the

checkword is placed in the EEPROM of microcontroller 32. Program step 78 waits for an indication that the crypto key will be transferred from key loader 22 to the EEPROM of microcontroller 32 with the crypto key being loaded into the EEPROM of microcontroller 32 during program step 82. Program step 80 loads the crypto key into the microcontroller 32. Microcontroller 32 and the computer software also duplicate the crypto key and check word in a backup location in the EEPROM of microcontroller 32.

Please replace the paragraph beginning at page 13, line 1 with the following paragraph:

The computer software next returns to main routine of FIG. 4. During program step 54, a determination is made as to whether or not the key should be erased. When the ERASE input to microcontroller 32 is high (RA4 input to microcontroller 32), the software proceeds to the erase key() routine 56 (FIGS. 4 and 8) and the microcontroller 32 erases the check word and the crypto key as well as its backup from the EEPROM within microcontroller 32. Five random writes are performed within the EEPROM within microcontroller 32. This logic one signal, i.e. ERASE signal is

provided by the loader interface 28 or the missile interface 30 to the RA4 input of microcontroller 32. The signal provided by the missile interface 30 is substantially higher than digital logic levels necessitating the use of additional resistor R9 in the LAUNCH line connecting missile interface 30 to microcontroller 32.

Please replace the paragraph beginning at page 13, line 16 with the following paragraph:

Referring to FIG. 8, the routine for erasing the EEPROM within microcontroller 32 is the erase_key routine 56. Program step 120 debounces the erase indication signal provided to the RA4 input to microcontroller 32. The computer software then proceeds to program step 122 to determine whether to erase the crypto key with its corresponding check word from the EEPROM within microcontroller 32. Whenever the signal provided to the RA4 input to microcontroller 32 is a logic "one", the computer software proceeds to program step 124 erasing the crypto key with its corresponding check word from the EEPROM within microcontroller 32. The erase light, i.e. light emitting diode 38 is set, and the load status is displayed during program step

124.

Please insert the following paragraphs into the specification beginning at page 14, line 2:

Referring to FIG. 9, FIG. 9 depicts the interrupt handler() routine 130. The interrupt handler() routine 130 looks for any interrupt within the computer software for microcontroller 32. For example, a timer interrupt will occur and a timer is initialized during program step 98 of FIG. 7. When the timer of program step 98 expires, an interrupt is generated and the software proceeds to the interrupt handler() routine 130. Program step 132 clears a timer interrupt flag and the software proceeds to program step 134, which is a decision program step. When the software determines that the crypto key and its associated checkword is to be loaded into encryption unit 24, the software proceeds to program step 136. During program step 136, the software continues the timer interrupt function and initializes transfer variables. The key load is also continued on the next interrupt.

Program step 138 is a decision program step which occurs during the loading of the crypto key and its associated checkword

into encryption unit 24. When the decision is "YES" the software proceeds to program step 140. During program step 140, the software continues the timer interrupt function and transitions the KGV clock. If the clock signal is high a new data bit is shifted out to encryption unit 24. When all the data has been shifted out of microcontroller 32 to encryption unit 24, the interrupt function is set for an end KGV key load. When the decision is "NO", the software proceeds to program step 142.

Program step 142 is a decision program step which determines if the KGV key load is completed. When the key end is completed, i.e. the answer is "YES", the software proceeds to program step 144 masking the timer interrupt and toggling the clock signal high. When the answer if "NO", the software proceeds to program step 146 which is a time delay decision program step.

When microcontroller 32 is in a time delay, the software proceeds to program step 150 reloading the timer, setting the timer for 10 millisecond intervals and if time has not expired setting the interrupt flag.

When there is no time delay, the software proceeds to program step 152. Program step 152 is a decision program step which is implemented when the ERASE LED 36 and the STATUS LED 38 are activated. Program step 154 reloads the timer and sets the

timer for indicator flash interval. In addition, the software toggles the indicator 36 or 38 if time has expired and also continues the timer interrupt function. Program step 156 is the return program step during which the software exits the interrupt handler() routine 130.